**Documento de Visão de Projeto**



**SHOPB**

**Histórico de Versões**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** | **Revisor** |
| 13/11/2022 | 1.0 | Modelagem e desenvolvimento | Leonardo e Jefferson | - |
| 19/11/2022 | 1.0 | Correção na formatação, Adicionados Casos de Uso e Visão de Dados, Correções nos Diagramas da camada de persistência e domínio | Leonardo e Jefferson | - |

|  |  |
| --- | --- |
| **Cliente** | FATEC - Interno |
| **Documento** | Documento de Visão de Projeto: *SHOPB* |
| **Data** | 13 de Novembro de 2022 |
| **Autores** | **Jefferson Akira Fukamizu**  akira.pitu@hotmail.com  **Leonardo Ferreira dos Santos**  leoferstos5@gmail.com |

**Página de Assinaturas**

|  |  |  |
| --- | --- | --- |
| Revisado e Aprovado por: |  |  |
|  |  | 13/11/21 |

**Índice**

Sumário

[1 Objetivo 4](#_Toc119755878)

[*1.1* *Escopo* 4](#_Toc119755879)

[*1.2 Referências* 4](#_Toc119755880)

[1.3 Necessidades de Negócio 4](#_Toc119755881)

[1.4 Objetivo do Projeto 4](#_Toc119755882)

[2 Declaração Preliminar de Escopo 5](#_Toc119755883)

[2.1 Descrição 5](#_Toc119755884)

[2.2 Produtos a serem entregues 5](#_Toc119755885)

[2.3 Requisitos 5](#_Toc119755886)

[Requisitos Funcionais 6](#_Toc119755887)

[Requisitos Não Funcionais 6](#_Toc119755888)

[Regras de Negócio 6](#_Toc119755889)

[3 Premissas 5](#_Toc119755890)

[4 Influência das Partes Interessadas 5](#_Toc119755891)

[5 Representação Arquitetural 5](#_Toc119755892)

[6 Visão de Use Case 7](#_Toc119755893)

[6.1 Diagrama de Caso de Fluxo Normal 8](#_Toc119755894)

[6.2 Descrição dos Casos de Uso Arquiteturalmente Significativos 8](#_Toc119755895)

[7 Visão de Lógica 10](#_Toc119755896)

[7.1 Camada de Apresentação 12](#_Toc119755897)

[7.2 Camada de Negócio 13](#_Toc119755898)

[7.2.1 Pacote Controller 14](#_Toc119755899)

[7.2.2 Pacote de Domínio 15](#_Toc119755900)

[7.5 Camada de Persistência 16](#_Toc119755901)

[7.6 Realização dos Casos de Uso Significativos 17](#_Toc119755902)

[8 Visão de Implantação 18](#_Toc119755903)

[9 Visão de Dados 18](#_Toc119755904)

[10 Tamanho e Performance 24](#_Toc119755905)

[11 Qualidade 25](#_Toc119755906)

[12 Cronograma Macro. 25](#_Toc119755907)

[13 Referências 26](#_Toc119755908)

# 1 Objetivo

Este documento trata principalmente da documentação das necessidades de negócios, da justificativa do projeto, do entendimento atual das necessidades do cliente e descreve resumidamente o novo produto, serviço ou resultado que deve satisfazer esses requisitos.

Tem o objetivo de alinhar as expectativas dos interessados para formalizar o início do projeto. Apresentar uma visão arquitetural do SHOPB. O intuito é salientar diferentes aspectos deste programa, obtidos a partir de decisões arquiteturais realizadas no âmbito dos sistemas de e-commerce.

## Escopo

O escopo deste documento trata do desenvolvimento de um módulo que atenda todas as necessidades de um e-commerce de jogos em mídia física.

O escopo deste documento é documentar as partes significativas do ponto de vista da arquitetura do modelo de design, como sua divisão em subsistemas e pacotes. Além disso, mostra sua divisão em classes e utilitários de classe.

## 1.2 Referências

Listar documentos relacionados a este documento de visão. Podem ser tanto documentos que foram utilizados para confeccionar este documento ou fazem parte deste documento, como documentos que estendem este documento.

Para a construção deste documento foram utilizadas as seguintes referências:

* Reuniões informais entre os desenvolvedores.
* Documentação disponibilizada pelo orientador.

# 2 Necessidades de Negócio

Um sistema que permita gerenciar as operações relacionadas a compras e produtos da loja de jogos SHOPB, cujas necessidades incluem.

- Gerenciar cadastro de Produtos.

- Gerenciar cadastro de Clientes.

- Gerenciar compras.

- Relatório geral de todas as vendas.

O sistema deve permitir controle estoque de produtos, com inativação automática caso o estoque acabe, além de permitir a trocas baseadas no valor do produto.

# 3 Objetivo do Projeto

Exemplo 1: Desenvolver uma plataforma para soluções web capaz de:

* armazenar informações em uma base de dados
* ser executado em qualquer navegador

Minimizando o tempo de atendimento dos alunos e professores que desejam emprestar algum livro do acervo e facilitando a produção de relatórios com base nos dados apresentados por tais operações.

# 4 Declaração Preliminar de Escopo

Esta seção descreve, em alto nível, o escopo do projeto. Os requisitos serão melhor detalhados nos documentos de Requisitos e Dicionário WBS.

## 4.1 Descrição

O produto do projeto é um e-commerce de jogos, responsável por realizar as vendas dos produtos do ramo. Ele é responsável pelo controle de estoque, acompanhamento de trocas de produtos vendidos, vendas, cadastro de clientes e relatórios sobre as vendas e produtos cadastrados.

## 4.2 Produtos a serem entregues

Os seguintes itens são considerados produtos do projeto, na sua etapa 1.

* Sistema do módulo da SHOPB , etapa 1, implementado de acordo com a especificação feita na fase de análise. (código objeto e código fonte).
* Documentos de especificação do sistema, concebido na fase de elaboração.
* Hospedagem do sistema em ambiente 24 x 7.

## 4.3 Requisitos

### 4.3.1 Requisitos Funcionais

* O sistema deve ser capaz de efetuar o cadastro, exclusão, alteração e consulta de jogos.
* O sistema deve ser capaz de efetuar o controle de trocas de produtos.
* O sistema deve ser capaz de produzir uma listagem com os jogos mais vendidos em determinado período.
* O sistema deve ser capaz de produzir relatórios referentes as operações do sistema.
* O sistema deve ser capaz de apresentar relatórios de vendas de todo o período de utilização do sistema.
* O sistema deve ser capaz de realizar a consulta a históricos.
* O sistema deve permitir a listagem das inativações dos produtos ao longo do tempo

### 4.3.2 Requisitos Não Funcionais

Utilizar o banco de dados PostGreSQL

A arquitetura da solução obedecer o design pattern MVC.

O sistema deve rodar nos seguintes browsers:

* Vivaldi
* Google Chrome
* Firefox

### 4.3.3 Regras de Negócio

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | | **Nome** | **Descrição** |
| **Grupo: Cadastro de Produtos** | | | |
| RN0001 | | Dados obrigatórios para o cadastro de um produto | Para todo produto cadastrado é obrigatório o cadastro dos seguintes dados: nome, descrição, preço, requisitos, publicadora, desenvolvedora, idioma, legenda, data de lançamento, imagem do produto |
| RN0002 | | Associação com categorias | Um produto pode estar associado com mais de uma categoria. |
| RN0003 | | Associar motivo de inativação | Todo produto que for inativado manualmente deve ter uma justificativa e uma categoria de inativação associada. |
| RN0004 | | Associar motivo de inativação automática | Todo cadastro de produto inativado de forma automática deve ser categorizado como FORA DE MERCADO. |
| RN0005 | | Associar motivo de ativação | Todo produto que for ativado deve ter uma justificativa e uma categoria de ativação associada. |
| **Grupo: Cadastro de Clientes** | | | |
| RN0006 | Cadastro de endereço de cobrança | | Para todo cliente cadastrado é obrigatório o registro de ao menos um endereço de cobrança. |
| RN0007 | Cadastro de endereço de entrega | | Para todo cliente cadastrado é obrigatório o registro de ao menos um endereço de entrega. |
| RN0008 | Composição do registro de endereços | | Todo cadastro de endereços associados a clientes deve ser composto dos seguintes dados: Logradouro, Número, Bairro, CEP, Cidade, Estado. Todos os campos anteriores são de preenchimento obrigatório. Opcionalmente pode ser preenchido um campo complemento e nome personalizado. |
| RN0009 | Composição do registro de cartões de crédito | | Todo cartão de crédito associado a um cliente deverá ser composto pelos seguintes campos: Nº do Cartão, Nome impresso no Cartão, Bandeira do Cartão e Código de Segurança. |
| RN0010 | Bandeiras permitidas para registro de cartões de crédito | | Todo cartão de crédito associado a um cliente deverá ser de alguma bandeira registrada no sistema. |
| RN0011 | Dados obrigatórios para o cadastro de um cliente | | Para todo cliente cadastrado é obrigatório o cadastro dos seguintes dados: Gênero, Nome, Data de Nascimento, CPF, Telefone (DDD e número), e-mail, senha |
| RN0012 | Validar retorno da operadora de cartão de crédito | | Somente deve-se dar baixa no estoque de itens cuja compra tenha sido efetivada, isso significa que o status não é mais EM PROCESSAMENTO. Todo item que faça parte de uma compra não aprovada deve ser desbloqueado e mantido em estoque. |
| **Grupo: Gerenciar Vendas Eletrônicas** | | | |
| RN0013 | Validar estoque para adição de itens no carrinho | | Não deve ser permitido adicionar um item no carrinho de compra que não esteja disponível em estoque. Também deve ser validado a quantidade do item adicionado ao carrinho para que não seja adicionado mais itens do que o disponível em estoque. |
| RN0014 | Uso de cupom promocional para pagamento | | Apenas um cupom promocional pode ser utilizado por compra. |
| RN0015 | Uso de diversos cartões de crédito | | Uma compra pode ser paga utilizando mais de um cartão de crédito, porém o valor mínimo para ser pago com cada cartão deve ser R$ 10,00. |
| RN0016 | Uso de cupons junto a cartão de crédito | | Ao realizar pagamento utilizando cupons e cartões em conjunto, deve-se sempre considerar o valor máximo dos cupons.  Somente neste caso é permitido que seja realizado um pagamento de um valor menor que R$ 10,00 no cartão. Exemplo: Uma compra de R$ 35,00 o cliente pode pagar R$ 30,00 utilizando cupons de troca ou cupons promocionais e pagar R$ 5,00 com cartão de crédito. |
| RN0017 | Gerar cupom de troca | | Um cupom de troca deve ser gerado quando uma compra for paga com outros cupons em que o valor supere o valor da compra. Obs: O sistema não deve possibilitar o uso de cupons que supere a compra desnecessariamente, ex: a venda tem valor total de R$ 50,00 e o cliente possui três cupons, um com valor de R$ 20,00, outro com valor de R$ 40,00 e um terceiro com valor de R$ 35,00 o sistema não deve possibilitar o uso dos três cupons nesta compra, deve ser aceito apenas dois cupons e consequentemente gerar um cupom com a diferença de R$ 5,00, ou R$ 10,00 ou R$ 25,00. |
| RN0018 | Validar Forma de Pagamento para finalização de compra | | Após a finalização da compra a forma de pagamento deve ser validada. Para tal deve-se validar a validade e veracidade dos cupons de troca e promocionais que porventura foram utilizados.  Também deve ser validado o aceite da compra pela respectiva operadora de cartão de crédito. |
| RN0019 | Alterar status da compra conforme processo de aprovação de forma de pagamento | | Caso as formas de pagamento tenham sido validadas com sucesso, a compra deve passar ter o status APROVADA.  Caso contrário deve passar a ter o status REPROVADA. |
| RN0020 | Alterar status da compra para transporte | | Toda compra selecionada para ser entregue por um administrador deve ter seu status alterado para EM TRANSPORTE. |
| RN0021 | Alterar status da compra após entrega | | Toda compra selecionada como entregue por um administrador deve ter seu status alterado para ENTREGUE. |
| RN0022 | Gerar pedido de troca | | Todo item selecionado para troca deve gerar um pedido de troca. Este pedido deverá terá o status EM TROCA.  Caso o cliente solicite a troca de toda a compra o status do pedido deverá ser EM TROCA. |
| RN0023 | Alterar status do pedido após recebimento de troca | | Ao confirmar que os itens de um pedido de troca ou uma compra com status EM TROCA foi recebido o status do pedido ou compra deverá ser TROCADO. |
| RN0024 | Validação para solicitar troca | | Somente itens de pedidos com status ENTREGUE poderão receber solicitação de troca. |
| RN0025 | Bloqueio de produtos | | Ao adicionar o item no carrinho, este deverá ser temporiamente bloqueado para que novas compras não sejam solicitadas. Tal bloqueio só deve ser retirado no caso da compra que gerou tal status não ser efetivada ou aprovada em um prazo parametrizado, o prazo deve levar em consideração o momento do bloqueio. Obs.: O prazo parametrizado deve ser relativo ao último item incluído no carrinho. |
| RNF0026 | Retirar item do carrinho | | Toda vez que um item for desbloqueado todos os itens do mesmo produto deverão ser retirados do carrinho de compra que gerou o prazo de bloqueio. |
| RNF0027 | Gerar notificação de autorização de troca | | Quando o administrador autorizar uma troca o sistema deverá gerar uma notificação sobre tal ao cliente. |
| **Grupo: Controle de estoque** | | | |
| RN0028 | Validar dados de estoque | | Para cada entrada em estoque, deve ser obrigatoriamente informado o produto, a quantidade, o valor de custo, fornecedor, e a data de entrada dos itens de produto. |
| RN0029 | Quantidade de itens | | Não deve ser permitido que seja realizado a entrada de itens de produtos com quantidade igual a zero. |
| RN0030 | Valor de custo | | Para todo item deve haver um valor de custo. |
| RNF0031 | Data de entrada | | Não deve ser permitido que itens sejam registrados sem que uma data de entrada seja registrada. |

# 5 Premissas

As premissas são informações que consideramos verdadeiras ou reais, e que vão sendo elaboradas gradativamente no decorrer do processo de Planejamento. Elas são documentas inicialmente no documento de visão e serão usadas como entradas em outros processos mais à frente.

* O projeto será orientado pelo professor Rodrigo Rocha.

# 6 Influência das Partes Interessadas

# 7 Representação Arquitetural

Os sistemas serão desenvolvidos tendo como base a arquitetura ilustrada na Figura 1. Toda a arquitetura será baseada nos padrões de projetos tradicionais do GoF e também nos padrões J2EE sendo executados dentro de um Servidor de Aplicações.

Diagrama

Descrição gerada automaticamente

**Figura 1 - Modelo Arquitetural Genérico**

O Container WEB abrigará os componentes da arquitetura responsáveis pela camada de apresentação. Será utilizado para esta camada o Reactjs uma livraria para utilização de Javascript na construção de páginas WEB.

As Classes de Domínio são as classes que representam os campos agrupados de acordo com a necessidade do negócio, contendo somente os atributos e os métodos e parâmetros para implementação das rn’s.

Nas classes de negócio o relacionamento com os DAOs responsáveis por persistir e recuperar os objetos no banco de dados, bem como gerenciar as regras de negócio.

Diagrama

Descrição gerada automaticamente

**Figura 2 - Diagrama em pacotes**

# 8 Visão de Use Case

Esta seção apresenta os Casos de Uso arquiteturalmente significativos, que foram selecionados considerando-se o pacote do Modelo de Casos de Uso que representa o sistema SHOPB.

A classificação dos casos de uso, em termos de significância, foi realizada com base na observação de pelo menos um dos seguintes critérios:

* Casos de uso que estendem outros Casos de Uso
* Casos de Uso que são incluídos em outros Casos de Uso e
* Casos de uso que acessam sistemas externos

## 8.1 Diagrama de Caso de Fluxo Normal

Diagrama

Descrição gerada automaticamente

Figura 8.1 Diagrama de Caso de Uso de Fluxo Normal

## 8.2 Descrição dos Casos de Uso Arquiteturalmente Significativos

**CLIENTE**

**Realizar cadastro**

Este caso de uso se inicia no primeiro acesso do cliente à plataforma, tendo que realizar um cadastro de conta para poder efetuar compras.

**Fazer login**

Este caso de uso se inicia após a finalização do cadastro, ou posteriores acessos do mesmo ao sistema.

**Gerenciar cadastro**

Após o login, o cliente pode gerenciar seu cadastro, como incluir, alterar ou excluir alguns dados do seu perfil.

**Consultar produtos**

Após o login, o cliente pode pesquisar por produtos que deseja na aba de pesquisa.

**Gerenciar compras**

Após o login, o cliente pode gerenciar suas compras, como editar os dados delas.

**\*Solicitar troca de produtos**

Este caso de uso se inicia opcionalmente após o cliente solicitar a troca de algum pedido recebido de uma compra que esteja finalizada.

**Gerenciar carrinho de compras**

Após o cliente adicionar itens ao seu carrinho de compra, ele pode incluir ou excluir itens do mesmo.

**Realizar Compra**

Após adicionar algum item ao carrinho de compra, o cliente pode confirmar a compra do pedido.

**\*Usar cupons**

Este caso de uso se inicia opcionalmente após o cliente escolher usar algum cupom de troca disponível na compra.

**ADMINISTRADOR**

**Fazer login**

Após registrar um login padrão, o administrador do sistema pode acessar o sistema.

**Consultar produtos**

Após o login, o administrador pode pesquisar por produtos que deseja na aba de pesquisa.

**Gerenciar Produtos**

O administrador, quando logado ao sistema, poderá cadastrar e alterar dados de produtos.

**\*Gerenciar estoque**

Após o login, o administrador pode gerenciar o estoque de produtos, como incluir, alterar ou excluir alguns produtos.

**Gerenciar compras**

Após o login, o administrador poderá gerenciar as vendas, editar seus dados e acompanhar o status delas.

**Gerenciar trocas**

Após o login, o administrador poderá gerenciar o campo trocas, onde irá autorizar ou negar pedidos de trocas e acompanhar trocas em andamento.

**\*Autorizar troca**

Este caso de uso se inicia opcionalmente quando o administrador autoriza pedidos de troca de produtos.

**\*Gerar cupom de troca**

Após autorizar a troca, o administrador deve gerar um cupom de troca para o cliente.

# 9 Visão de Lógica

Esta visão apresenta elementos de design significativos do ponto de vista da arquitetura, descrevendo a organização do Sistema SHOPB em pacotes, bem como a organização desses pacotes em camadas.

O Diagrama com as camadas do sistema SHOPB é ilustrado na figura 5.1.

Diagrama

Descrição gerada automaticamente

Figura 5.1 – Diagrama de camadas do Sistema

**Apresentação**: Contém classes para as interfaces gráficas com os usuários (GUI). Através destas interfaces os usuários conseguem interagir com a SHOPB, com o intuito de incluir, alterar e excluir produtos.

**Negócio**: Contém classes que controlam a execução das funcionalidades da SHOB.

**Persistência**: Contém classes responsáveis por persistir as entidades de domínio. Por exemplo,

contém as classes que permitem ler e gravar os objetos no banco de dados relacional.

A Figura 5.2 ilustra o diagrama de camadas com as tecnologias utilizadas no desenvolvimento, já descritas na figura 5.1.

Diagrama

Descrição gerada automaticamente

Figura 5.2: Camadas do shopb com as dependências de tecnologia

## 9.1 Camada de Apresentação

Nesta camada, temos o pacote viewhelper que contém as classes relacionadas à exibição e recebimento de informações pelo usuário através da visualização. A classe controller será a responsável por gerenciar as requisições do sistema.

Diagrama

Descrição gerada automaticamente

Figura 5.3: Camada de Apresentação

## 9.2 Camada de Negócio

Nesta camada, temos o pacote controle que contém as classes responsáveis por controlar as regras de negócio da aplicação. O pacote model, contém as classes que representam o modelo, ou seja, aquelas que contém as informações sobre o Sistema SHOPB. O pacote controller contém as classes de controle do negócio. O pacote complementador contém classes que auxiliam na complementação de informações e o pacote filtro contém as classes que auxiliam na filtragem de informações.

A figura 5.4 ilustra os pacotes descritos.

Diagrama, Esquemático

Descrição gerada automaticamente

Figura 5.4: Camada de Negócios

### 9.2.1 Pacote Controller

A figura 5.5 ilustra as principais classes de controle.

Diagrama

Descrição gerada automaticamente

Figura 5.5: Classes de controle

### 9.2.2 Pacote de Domínio

A figura 5.6. ilustra as principais classes do modelo.

Diagrama

Descrição gerada automaticamente

Figura 5.6: Classes de Domínio

## 9.5 Camada de Persistência

Nesta camada temos o pacote DAO que contém as classes e interfaces responsáveis por persistir as informações da SHOPB no BD relacional. A livraria typeorm é utilizada nas classes DAO e algumas de suas interfaces são reaproveitadas nos métodos dos DAO’s. Todos os DAO’s implementam o CRUD básico a partir da classe abstract DAO, apenas no caso do produto foram necessárias algumas implementações especificas para gerar o dashboard do administrador e implementar a regra de produtos reservados do carrinho do cliente

Diagrama

Descrição gerada automaticamente

Figura 5.7: Camada de Persistência.

Diagrama

Descrição gerada automaticamente

Figura 5.8 Classes do Pacote DAO

## 9.6 Realização dos Casos de Uso Significativos

Diagrama

Descrição gerada automaticamente

Figura 5.9 – Caso de Cadastro de Usuario Cliente

Diagrama

Descrição gerada automaticamente

Figura 5.10 – Caso de Listagem de Produtos

# 10 Visão de Implantação

Esta seção descreve as configurações da rede física (hardware) na qual o SHOPB será implantado e executado.

Trata-se de uma visão do Modelo de Implantação que, para a configuração em questão, indica os nós físicos (computadores, CPUs), que executarão o subsistema SHOPB, e as respectivas interconexões (barramento, LAN, etc). A figura 6 ilustra o modelo de implantação para o SHOPB.

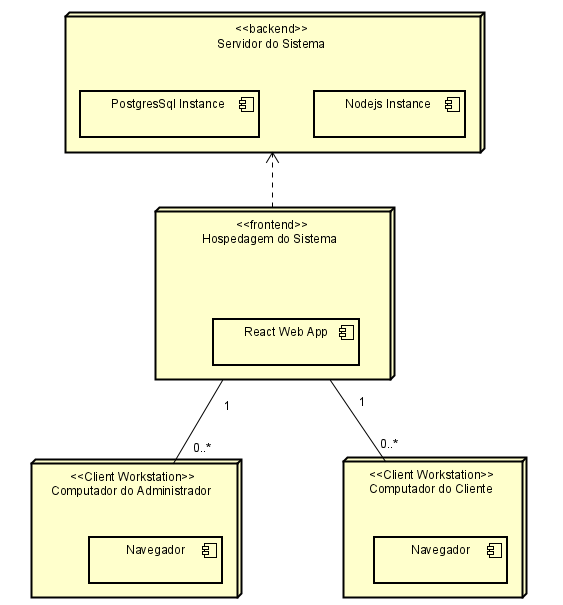


Figura 6: Visão de Implantação do SHOPB

# 11 Visão de Dados

O mecanismo de persistência utilizado no sistema SHOPB utiliza-se o banco de dados Relacional Postgres juntamente com a livraria para mapeamento objeto-relacional, typeorm.

As figuras 11.1 e 11.2, apresentam a visão lógica e física da base de dados do SHOPB.

Diagrama

Descrição gerada automaticamente

Figura 11.1 – Modelo Lógico

Diagrama

Descrição gerada automaticamente

Figura 11.2 – Modelo Físico

A Tabela 2 define o mapeamento das principais classes de modelo para entidades do modelo lógico do BD Oracle.

Note que existem alguns campos nas entidades lógicas do BD que não estão mapeadas diretamente com as classes de modelo da Visão Lógica contidas neste documento:

Os campos que começam com “IN” são campos definidos através de domínios do BD. As classes de modelo que possuem os atributos que representam tais campos não constam neste documento, mas podem ser consultadas no documento [REF – Modelo de Design].

Os campos #NRSQ e DT REGISTRO presentes em todas as entidades lógicas do BD, correspondem a atributos presentes em superclasses das classes de modelo, que não constam neste documento, mas podem ser consultadas no documento [REF – Modelo de Design].

Tabela 2: Mapeamento Objeto-Relacional

|  |  |
| --- | --- |
| **Classe** | **Entidade** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# 12 Tamanho e Performance

O sistema SHOPB será usado para o controle do comércio exterior e conseqüentemente terá uma grande base.

Seus servidores provavelmente irão passar por períodos de picos de utilização (por exemplo, próximo a finalização de prazos de entrega de documentos e/ou procedimentos fiscais, tal como ocorre com os servidores que recebem declarações de imposto de renda).

As estimativas do número de usuários e de carga de utilização em períodos de pico de utilização, bem como maiores informações sobre questões relacionadas ao tamanho e desempenho do sistema SHOPB podem ser obtidas no documento de requisitos não funcionais.

# 13 Qualidade

O sistema SHOPB será usado para o controle do comércio exterior brasileiro, conseqüentemente tratando de altos volumes financeiros e um grande número de operações de importação/ exportação diariamente.

Eventuais erros e/ou falhas na sua operação podem levar a prejuízos significativos tanto em termos financeiros quanto na imagem da UMC, portanto na fase de design deve-se levar em consideração como fatores prioritários a confiabilidade e robustez do sistema.

Adicionalmente, o sistema SHOPB pode ser alvo de ataques de “hackers” para roubar ou simplesmente corromper informações, possibilidade aumentada pela interface do sistema disponível na Internet, para evitar que tais ataques sejam bem sucedidos uma infra-estrutura de segurança deve ser especificada e projetada.

Maiores informações sobre questões relacionadas aos requisitos de qualidade do sistema SHOPB podem ser obtidas no documento de requisitos não funcionais.

# 14 Cronograma Macro.

Os prazos podem ser dados em semanas ou meses dentro do projeto ou ate mesmo data.

|  |  |
| --- | --- |
| **Resultado** |  |

# 15 Referências

Unified Modeling Language: <http://www.omg.org/technology/documents/formal/uml.htm>

RUP. Rational Unified Process.